

SAP Community > Products and Technology > Technology

> Technology Blogs by Members > Alternative to Fiori and oData, Vue and Rest Api

## Alternative to Fiori and oData, Vue and Rest Api



**beyhan\_meyrali**  
Active Contributor

2022 Dec 23 11:21 AM



8 Kudos

2,982 Views

Hi Community,

Since we can do UI5 development with VS Code and deploy our web app to BSP stack on SAP Netweaver. We can do exactly same with any other javascript framework.

Personally I am not fun of oData therefore I try alternative ways of building web apps which are hosted on SAP.

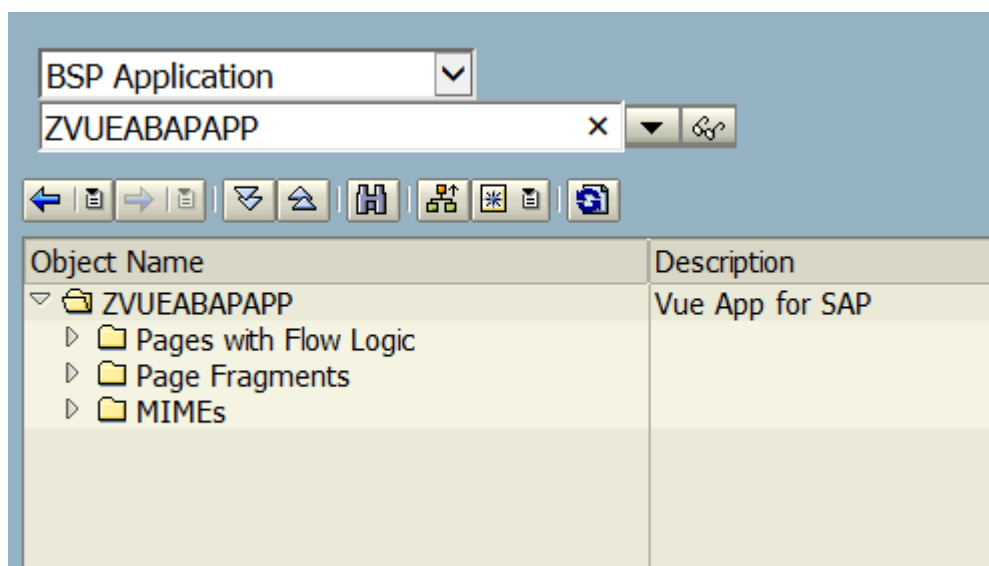
In my first blog/video tutorial I have shown, how to build a Vue.js app with VS Code and How to publish it to BSP stack. Tutorial is below.

And then, in another blog/video tutorial, I have shown how to create rest api with SAP Abap and how to test it on browser and Postman.

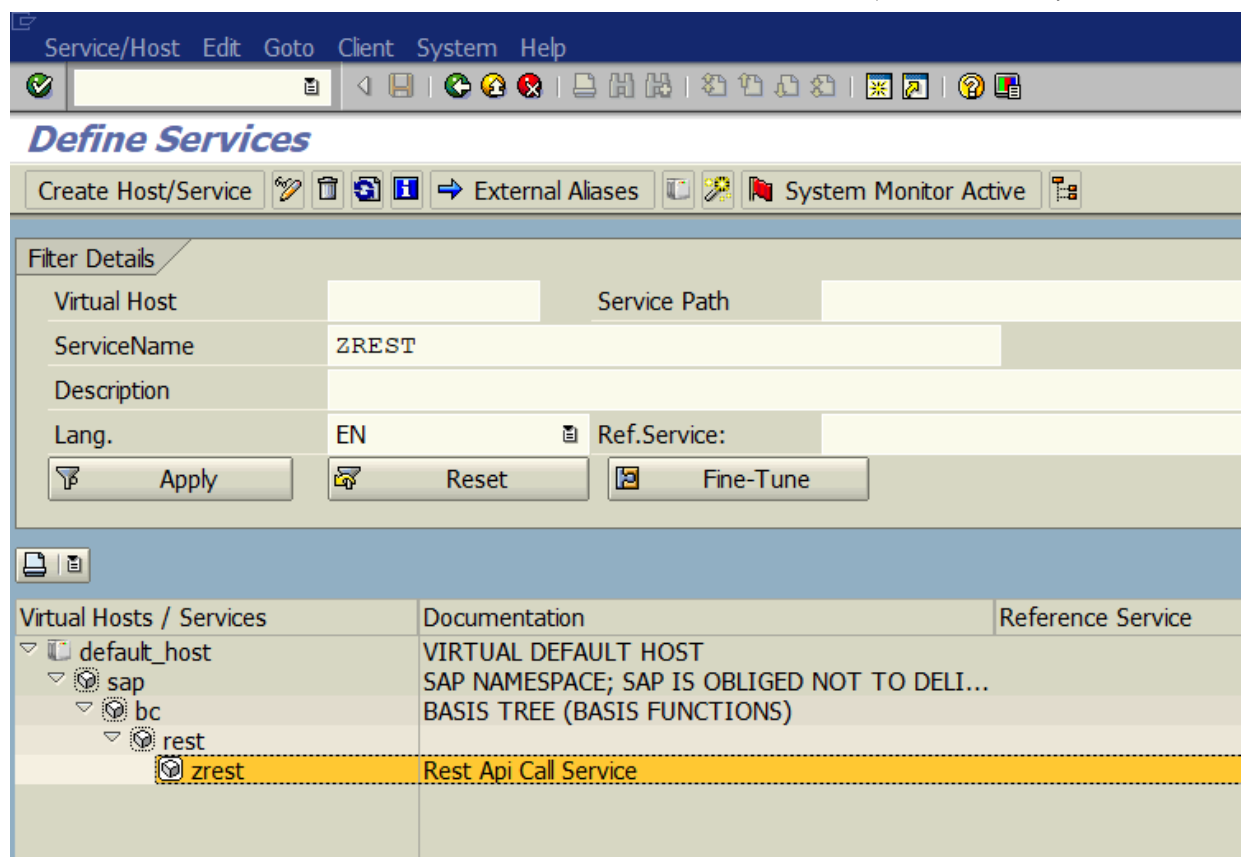
And this final blog of Vue & Rest Api series, I will show you how to consume that rest api.



If you have done steps in those two tutorials, now you should have a vue app on SAP and also a rest service.



*Vue App on BSP Stack*



### Rest Api Service

If you have them both working. Now we just need to make call to that rest api from Vue app and publish it again.

Simply open VS Code and change content of HelloWorld.vue with code below to make a rest api call to SAP by using axios library.

```
<template>
  <div id="restapp">
    <h1>{{ msg }}</h1>
    <div>
      <div v-if="loading">
        loading...
      </div>
      <div v-else>
        <div>
          <p>Date Time : {{ rObj.DATETIME }}</p>
        </div>

        <div>
          <p>Response : {{ response }}</p>
        </div>

        <div>
          <button @click="getDateTime">Refresh</button>
        </div>
      </div>
    </div>

    <h1>Test Rest Api</h1>
    <p><a target="_blank"
      href="http://vhcalnplci:8000/sap/bc/rest/zrest/Rest?sap-client=800"
      to Test Rest Api</a></p>

  </div>
</template>

<script>
const URL = 'http://vhcalnplci:8000/sap/bc/rest/zrest/Rest';

import axios from 'axios';

export default {

  name: 'HelloWorld',
  props: {
    msg: String
  },
}
```

```
el: '#restapp',
data() {
  return {
    loading: true,
    posts: [] // add posts here so reactivity is working, also unde
  }
},
methods: {
  getDateTime() {
    this.loading = true
    axios.get(URL, {
      params: {
        req: '{"ID":1001,"BODY":"Some Json Content"}'
      }
    }).then((response) => {
      console.log(response.data, this)
      this.response = response.data
      this.rObj = JSON.parse(response.data.BODY);

      this.loading = false
    })
  },
},
created() {
  this.getDateTime()
}
}

</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
h3 {
  margin: 40px 0 0;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
```

```

display: inline-block;
margin: 0 10px;
}

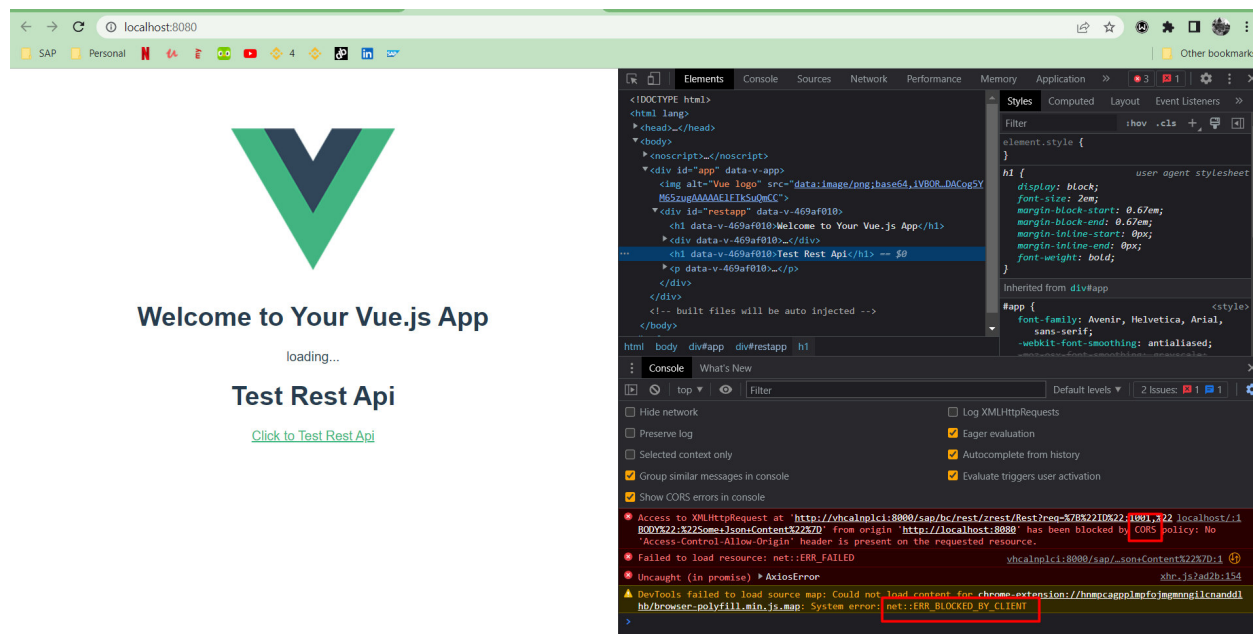
a {
  color: #42b983;
}
</style>

```

run your server with "**npm run serve**" if it is not already running.

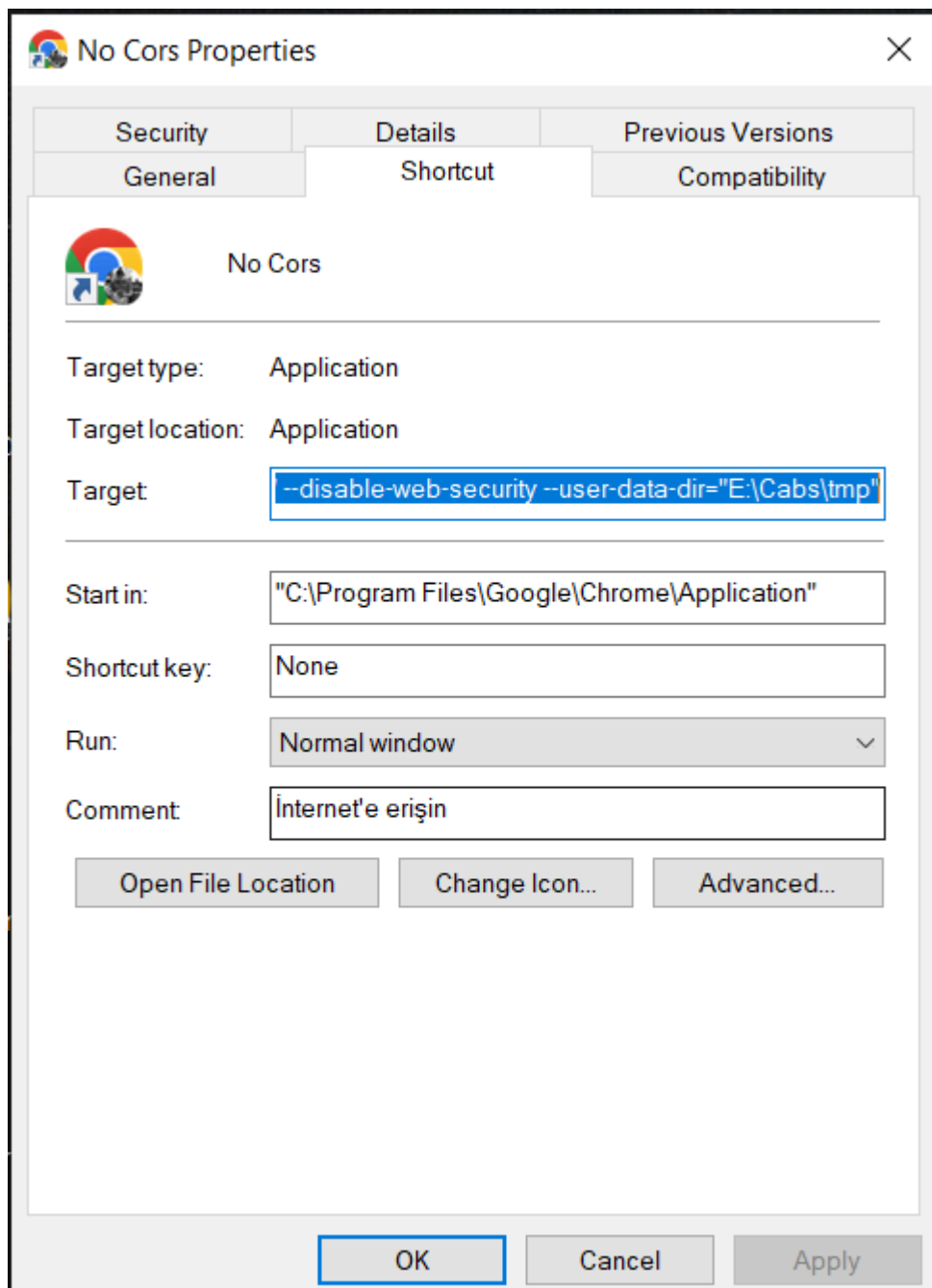
Code calls created method when vue object is created on page load and makes rest api call to SAP Rest api url with get parameters. Get parameters are handled in zrest api service, as told in detail in rest api tutorial, and result is returned as json object as seen below.

To test app on localhost you need to cancel CORS on Google Chrome, if you are too using Chrome.



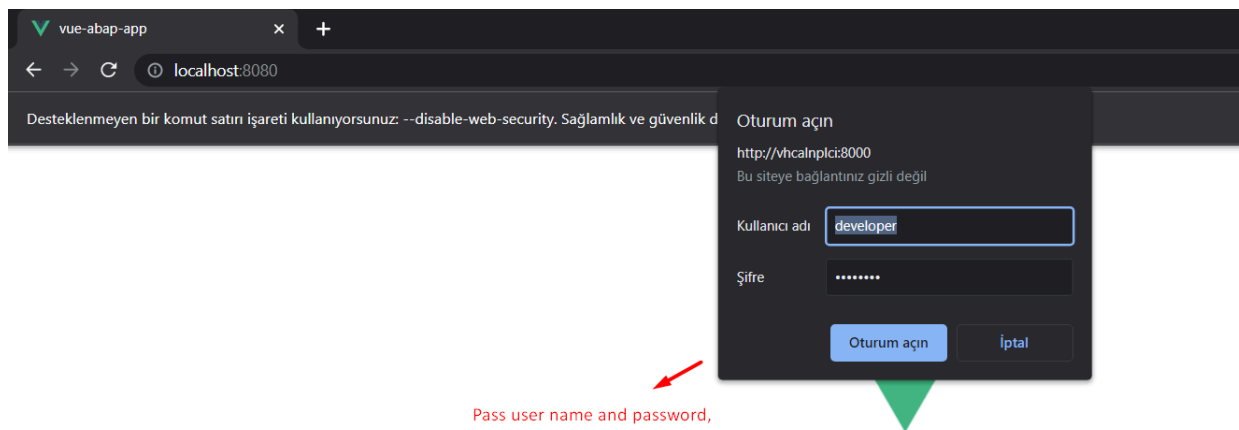
### Cors Error!

You can change the target of Google Chrome shortcut -> "C:\Program Files\Google\Chrome\Application\chrome.exe" --**disable-web-security** --user-data-dir="E:\Cabs\tmp" instead of E:\Cabs\tmp you can give any directory.



### *Disable Cors on Chrome*

And if you try to open chrome with that new link and enter our url you will get the data from SAP.



## Welcome to Your Vue.js App

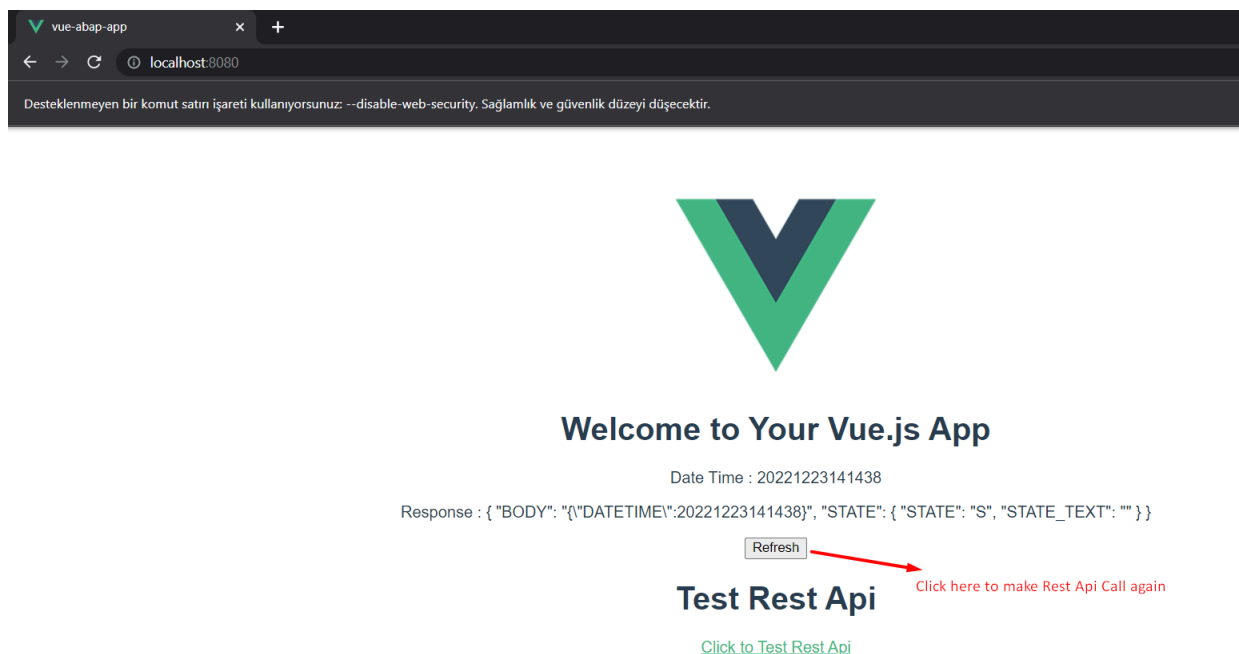
loading...

### Test Rest Api

[Click to Test Rest Api](#)

*Enter credentials to logon to SAP system*

And the result!



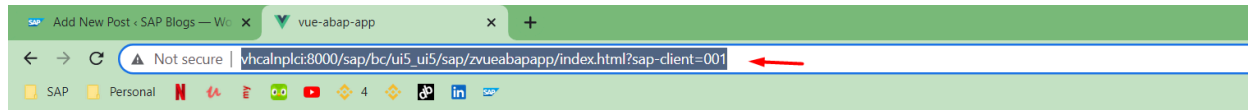
If you publish it to SAP and test it from sap url, you will have no Cors issues.



To publish first build your app with "**npm run build**" and publish it to SAP with "**npx nwabap upload**".

Now we can open normal chrome and we can enter our app url ->

[http://vhcalnplci:8000/sap/bc/ui5\\_ui5/sap/zvueabapapp/index.html?sap-client=001](http://vhcalnplci:8000/sap/bc/ui5_ui5/sap/zvueabapapp/index.html?sap-client=001)



## Welcome to Your Vue.js App

Date Time : 20221223141811

Response : { "BODY": "{ \"DATETIME\": 20221223141811 }", "STATE": { "STATE": "S", "STATE\_TEXT": "" } }

[Refresh](#)

## Test Rest Api

[Click to Test Rest Api](#)

Exactly the same result 😊

That is it. Rest of development is pure Vue on frontend and pure abap on backend. Very simply this how you can use any javascript framework on BSP stack with Rest api backend.

Now, instead of Fiori and oData, you can use Vue and Rest Api.

Hope that helps you.

Thanks for reading.

### Tags:

JSON

REST API

vue

vue.js

### Comments



dukejib5 Participant

2023 Jan 19

7:58 AM



I was literally thinking of using Vue/Nuxt for Fiori Development and you actually came up with this blog.

That's fantastic and very useful.

Will read your 2 related blogs as well and implement it on test server 😊



beyhan\_meyrali Active Contributor

2023 Jan 19

11:04 AM

Hi Ali,

Thanks for comment 😊

Glad, you found those useful.



beyhan\_meyrali Active Contributor

2024 May 15

7:19 AM

-

edited

2024 May 15

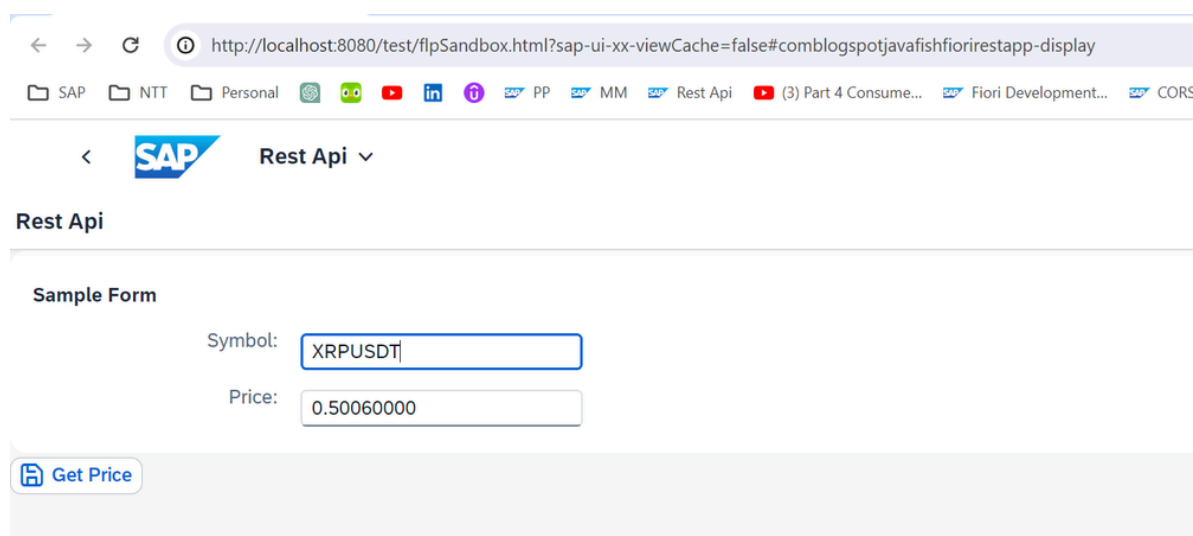
7:28 AM

- edited

Here is another example of Rest Call usage in Fiori.

In this example, we are making a rest get call to Binance to get current ticker price.

To learn how to create a Fiori SAP UI5 app in VSCode, [you can read that blog.post](#).



To avoid CORS errors, install cors-anywhere

```
npm i cors-anywhere
```

To run multiple scripts, install npm run all

```
npm install -g npm-run-all
```

Controller Code



```

sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/model/json/JSONModel",
    "sap/m/MessageToast"
],
    /**
     * {typeof sap.ui.core.mvc.Controller} Controller
     */
    function (Controller, JSONModel, MsgBox) {
        "use strict";

        return Controller.extend("com.blogspot.javafish.fioriextension")
            .onInit: function () {
                this.fetchData("BTCUSDT");
            },
            fetchData: function (symbol) {
                var that = this;
                $.ajax({
                    type: "GET",
                    url: "http://localhost:8081/https://api.binance.com/api/v3/ticker/price?symbol=" + symbol,
                    contentType: "application/json",
                    dataType: "json",
                    success: function (respData) {
                        that.setViewData(respData);
                    }
                }).always(function (data, status, response) {
                    //MsgBox.show("Data:" + data + " Status:" + status);
                });
            },
            getPriceForSymbol: function () {
                var modelData = this.getView().getModel().getData();
                let sym = "" + modelData.payload.symbol;
                sym = sym.toUpperCase();
                this.fetchData(sym);
            },
            setViewData: function (data) {
                var oModel = new JSONModel();
                oModel.setData({
                    "payload": data
                });
                this.getView().setModel(oModel);
            }
    }
);

```

```
    });  
  });
```

## View File Code

```
<mvc:View xmlns:core="sap.ui.core" xmlns:form="sap.ui.layout.form"  
  xmlns:mvc="sap.ui.core.mvc" displayBlock="true"  
  xmlns="sap.m">  
  <Page id="page" title="{i18n>title}">  
    <content>  
      <form:SimpleForm id="_IDGenSimpleForm1" xmlns:sap.ui.layo  
      <form:content>  
        <core:Title text="Sample Form" id="lblHead"/>  
        <Label text="Symbol" id="lblSymbol"></Label>  
        <Input width="200px" id="txtSymbol" value="{/payload/  
        <Label text="Price" id="lblPrice"></Label>  
        <Input width="200px" id="txtPrice" value="{/payload/p  
      </form:content>  
    </form:SimpleForm>  
    <Button text="Get Price" icon="sap-icon://save" press="ge  
  </content>  
</Page>  
</mvc:View>
```

## Cors-anywhere Proxy settings

```
// Listen on a specific host via the HOST environment variable
var host = process.env.HOST || '0.0.0.0';
// Listen on a specific port via the PORT environment variable
var port = process.env.PORT || 8081;

var cors_proxy = require('cors-anywhere');
cors_proxy.createServer({
  originWhitelist: [], // Allow all origins
  requireHeader: ['Origin', '*'],
  removeHeaders: ['cookie', 'cookie2']
}).listen(port, host, function() {
  console.log('Running CORS Anywhere on ' + host + ':' + port);
});
```

## Package Json

```
{
  "name": "fiorirestapp",
  "version": "0.0.1",
  "private": true,
  "description": "A Fiori application.",
  "keywords": [
    "ui5",
    "openui5",
    "sapui5"
  ],
  "main": "webapp/index.html",
  "devDependencies": {
    "@sap/ux-ui5-tooling": "1",
    "@ui5/cli": "^3.10.1",
    "cors-anywhere": "^0.4.3"
  },
  "scripts": {
    "start": "fiori run --open \"test/flpSandbox.html?sap-ui-xx-v\"",
    "start-local": "fiori run --config ./ui5-local.yaml --open \"\"",
    "build": "ui5 build --config=ui5.yaml --clean-dest --dest dis",
    "deploy": "fiori verify",
    "deploy-config": "fiori add deploy-config",
    "start-noflp": "fiori run --open \"index.html?sap-ui-xx-viewC",
    "start-variants-management": "fiori run --open \"preview.html",
    "unit-tests": "fiori run --open 'test/unit/unitTests.qunit.ht",
    "int-tests": "fiori run --open 'test/integration/opaTests.qun",
    "proxy": "node proxy.js",
    "startall": "npm-run-all --parallel start proxy"
  },
  "ui5": {
    "dependencies": [
      "@sap/ux-ui5-tooling"
    ]
  },
  "sapuxLayer": "CUSTOMER_BASE",
  "dependencies": {
    "cors-anywhere": "^0.4.4"
  }
}
```



To run the app, use start all script

```
npm run startall
```

### Related Links

<https://blog.surges.eu/cors-proxy-server-for-remote-odata-service-in-local-sapui5-dev/>

<https://www.npmjs.com/package/cors-anywhere>

I wonder, why SAP insist on oData. While we can create very lean solutions with rest.

Hope this helps.

Regards



greenkernel Discoverer

a week ago

because large enterprise does not want to implement frameworks that has updates overnight. It's a chaos.

Powered by

